



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/747,063	12/22/2000	Timothy A. Best	STL919990186US2/A8063	1655

46159 7590 01/17/2012
SUGHRUE MION PLLC
USPTO CUSTOMER NO WITH IBM/SVL
2100 PENNSYLVANIA AVENUE, N.W.
WASHINGTON, DC 20037

EXAMINER

PILLAI, NAMITHA

ART UNIT	PAPER NUMBER
----------	--------------

2172

MAIL DATE	DELIVERY MODE
-----------	---------------

01/17/2012

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte TIMOTHY A. BEST, MICHAEL J. DOCKTER,
STEPHAN GRELL, and RONALD W. LYNN

Appeal 2009-013220
Application 09/747,063
Technology Center 2100

Before DENISE M. POTHIER, MICHAEL R. ZECHER, and
BRUCE R. WINSOR, *Administrative Patent Judges*.

POTHIER, *Administrative Patent Judge*.

DECISION ON APPEAL

Appellants appeal under 35 U.S.C. § 134(a) from the Examiner's rejection of claims 1-42, 46-48, 51, 52, and 54-60. Claims 43-45, 49, 50, and 53 have been canceled. App. Br. 4.¹ We have jurisdiction under 35 U.S.C. § 6(b). We affirm-in-part.

STATEMENT OF THE CASE

Appellants' invention relates to a technique for delivering multiple applets within a fixed-size viewing space. *See* Spec. 1:10-11. Claim 1 is reproduced below with the key disputed limitation emphasized:

1. A method of executing applets, comprising:
 receiving a user selection of a plurality of applets;
 generating separate windows within a main applet for each selected applet;
 displaying the main applet; and
 the main applet executing each selected applet in a separate window, *wherein each selected applet is displayed in the separate window inside the displayed main applet.*

The Examiner relies on the following as evidence of unpatentability:

Southgate	US 5,561,757	Oct. 1, 1996
Banthia	WO 98/43170	Oct. 1, 1998

Sun Microsystems, Inc., *The Swing Tool Set*, 3 SWING CONNECTION 1-13² (1998), *available at* http://www.cs.bris.ac.uk/~worrall/tool_set.html ("Sun").

THE REJECTIONS

1. The Examiner rejected claim 52 under 35 U.S.C. § 112, ¶ 1, as failing to comply with the written description requirement. Ans. 3.³

¹ Throughout this opinion, we refer to (1) the Appeal Brief filed December 13, 2007; (2) the Examiner's Answer mailed March 4, 2008; and (3) the Reply Brief filed May 5, 2008.

² Thirteen printed pages of this reference were provided, and these page numbers correspond sequentially to the pages provided.

2. The Examiner rejected claims 1-4, 9-12, 14-18, 23-26, 28-32, 37-40, 42, 46-48, 51, 52, 54-58, and 60 under 35 U.S.C. § 103(a) as unpatentable over Banthia. Ans. 3-10.

3. The Examiner rejected claims 5-8, 19-22, 33-36, and 59 under 35 U.S.C. § 103(a) as unpatentable over Banthia and Southgate. Ans. 10-11.

4. The Examiner rejected claims 13, 27, and 41 under 35 U.S.C. § 103(a) as unpatentable over Banthia and Sun. Ans. 11.

THE LACK OF WRITTEN DESCRIPTION REJECTION

Regarding claim 52, the Examiner finds that the Specification does not describe the recitation, “at least two of said plurality of applets do not inherit functions from [the] same base class” in such a way as to reasonably convey to an ordinary skilled artisan that Appellants were in possession of this recited subject matter. Ans. 3, 11-12. Appellants argue that the Specification supports such a limitation. App. Br. 13-15; Reply Br. 4-6. Appellants assert that the disclosure discusses various independent applets (e.g., “sevenAMNewsTicker.class” and “MyApplet.class”) and that one skilled in the art would have understood that these applets do not inherit functions of the same base class. App. Br. 13-14. Appellants also contend the disclosure of applets having different functions leads an ordinary artisan to understand that such applets are “less likely . . . to inherit functions from the same base class.” App. Br. 14.

³ The 35 U.S.C. § 112, ¶ 1, rejection of claims 57-60 has been withdrawn. Ans. 2.

ISSUE

Under § 112, first paragraph, has the Examiner erred in rejecting claim 52 by finding Appellants' disclosure does not reasonably convey to an ordinary skilled artisan that Appellants were in possession of "at least two of said plurality of applets do not inherit functions from [the] same base class"?

ANALYSIS

Based on the record before us, we find no error in the Examiner's rejection of claim 52. First, Appellants do not use the words, "inherit," "base class," or a derivative of these phrases, in their disclosure as originally submitted. *See generally* Specification. Second, contrary to Appellants' assertion (App. Br. 13), Appellants only state that multiple applets are used within a browser space and do not describe the applets' functions as independent of each other. *See* Spec. 7:10-15. Also, Appellants do not describe a loan calculator or word processor applet in the disclosure. *See* App. Br. 14.

Moreover, we do not find that applets having different names (e.g., "Did You Know?," "Corporate News," "Business Intelligence" (Spec. 11:17-22; Fig. 4)) reasonably convey to an ordinary skilled artisan that Appellants are in possession of subject matter of the applets not inheriting functions from the same base class. For example, Appellants describe code for two applets embedded in a web page (e.g., code for "sevenAMNewsTicker.class" and code for "MyApplet.class"). Spec. 7:22-8:26. Arguably, one skilled in the art would have understood applets having different names coupled with the extension, "class," as different from each other and even being from a different class. *See id.* Nonetheless, the

extension, “class,” is not labeled as or indicative of a base class.

“SevenAMNewsTicker.class” and “MyApplet.class” may describe subclasses that both branch from a given base class, and thus, both these applets may inherent functions from the same base class.

Nor have Appellants provided adequate evidence that such “class” in the code is a “base class” as recited. *See* App. Br. 13-15. For the first time in the Reply Brief, Appellants argue that “when two applets carry out different types of functionalities, they would belong to different JAVA base classes.” Reply Br. 4. Such an argument is untimely and waived. *See Ex parte Borden*, 93 USPQ2d 1473, 1474 (BPAI 2010) (informative) (“[T]he reply brief [is not] an opportunity to make arguments that could have been made in the principal brief on appeal to rebut the Examiner’s rejections, but were not.”). Also, other than pointing out that the applets have different names, this assertion is unsupported by factual evidence and is entitled to little probative value. *See In re Geisler*, 116 F.3d 1465, 1470 (Fed. Cir. 1997).

We therefore find that the disclosure does not reasonably convey to one skilled in the art that they had possession of the limitation, “at least two of said plurality of applets do not inherit functions from [the] same base class,” as of the time of filing. For the foregoing reasons, Appellants have not persuaded us of error in rejecting independent claim 52.

THE OBVIOUSNESS REJECTION OVER BANTHIA

Contentions

Regarding representative claim 1, the Examiner finds that, while Banthia teaches that the controlling applet has a height and width of zero or

is hidden, Banthia further suggests that the height and width of the applet can be altered, such that the displayed applets are displayed in separate windows inside the controlling applet. Ans. 4, 12-13. Appellants argue that Banthia fails to teach or suggest displaying each selected applet in a separate window inside the displayed main applet as recited because the controlling applet is hidden and the displayed applets are displayed within a browser window. App. Br. 19, 21-22; Reply Br. 6-8. Appellants further assert that the Examiner has not provided a reason for displaying separate applets inside the controlling or main applet. App. Br. 20.

As for claim 11, Appellants assert that Banthia does not teach or suggest the controlling applet being a webtop applet. App. Br. 24; Reply Br. 10.

As for claim 12, Appellants contend Banthia does not teach or suggest having an available applets list being displayed within the main applet. App. Br. 24; Reply Br. 10.

As for claim 52, Appellants argue that Banthia teaches that the displayed applets (a) inherit from the same base class, which is contrary to the recite claim language, and (b) are interdependent. App. Br. 24-25; Reply Br. 11.

Regarding claim 54, Appellants assert that tearing off applets does not teach receiving a user selection of applets at substantially the same time. App. Br. 25; Reply Br. 11-13.

Regarding claim 57, the Examiner finds that Figure 5 teaches “selecting a plurality of applets from a displayed list of the plurality of available applets.” Ans. 8. The Examiner further explains that because the information in the applets is updated, the displayed applets have not been

uploaded until the information is added and redisplayed. Ans. 9, 13.

Appellants contend that Banthia teaches that the displayed applets are loaded rather than are applets not yet loaded for execution as recited. App. Br. 23-24; Reply Br. 8-9.

ISSUES

(1) Under § 103, has the Examiner erred in rejecting claim 1 by finding that Banthia would have taught or suggested “each selected applet is displayed in the separate window inside the displayed main applet”?

(2) Under § 103, has the Examiner erred in rejecting claim 11 by finding that Banthia would have taught or suggested the main applet is a webtop applet?

(3) Under § 103, has the Examiner erred in rejecting claim 12 by finding that Banthia would have taught or suggested executing the main applet to display within the main applet an available applets list from which the user can select applets?

(4) Under § 103, has the Examiner erred in rejecting claim 52 by finding that Banthia would have taught or suggested the applets are independent and at least two applets do not inherit functions from the same base class?

(5) Under § 103, has the Examiner erred in rejecting claim 54 by finding that Banthia would have taught or suggested receiving a user selection of the applets at substantially the same time?

(6) Under § 103, has the Examiner erred in rejecting claim 57 by finding that Banthia would have taught or suggested selecting from a

displayed list of available applets displayed in the main applet, wherein the applets displayed in the list are not yet loaded for execution?

FINDINGS OF FACT (FF)

Banthia

1. Banthia states, “[t]ypically, the controlling applet is a hidden applet. This means that its height and width parameters are zero.” 5:33-35.
2. Banthia’s Figure 5 displays six smaller display applets inside a larger browser window. 4:36-38; 5:36–6:2; 8:19-21; Fig. 5.
3. Banthia’s controlling applet interfaces and communicates with the display applets. The controlling applet maintains a connection to the server and continues to update each “torn off” applet with new data. 5:29-31; 8:34-37.
4. Banthia teaches a user can move a display applet to another window, such as clicking on the “Window” icon corresponding to Packet Size Distribution. The “torn off” applet window will move to another screen location of the user’s display screen (e.g., upper top left corner). 8:25-33.

ANALYSIS

Claims 1-4, 9, 10, 14-18, 23-36, 28-32, 37-40, 42, 46-48, 51, 55, and 56

Based on the record before us, we find no error in the Examiner’s rejection of representative claim 1 which calls for, “each selected applet is displayed in the separate window inside the displayed main applet.” As the Examiner explains (Ans. 4, 12), Banthia states, “[t]ypically, the controlling applet is a hidden applet.” FF 1. However, as the phrase, “typically,” suggests, Banthia suggests scenarios where the controlling applet can be

visible or in atypical situations. We therefore agree with the Examiner (Ans. 4, 12) that Banthia suggests the controlling applet can be visible or displaying a main applet as recited by adjusting the height and width parameters to something greater than zero. *See* FF 1.

Additionally, contrary to Appellants' assertions (App. Br. 20-21; Reply Br. 8), this teaching of making the controlling applet visible in atypical situations does not teach away from displaying the controlling applet. That is, while Banthia's controlling applet is *typically* hidden, this teaching does not discourage an ordinary skilled artisan from making the applet visible to the user or that there is no need to display the controlling applet. Rather, Banthia teaches or suggests an alternative to hiding the applet. Also, the "tearing off" feature in Banthia does not teach away (App. Br. 20; Reply Br. 7) from displaying the separate applets within a main applet. In fact, this "tearing off" feature suggests a technique for placing applets at various locations, including inside other windows or applets. *See* FF 4. An ordinary skilled artisan would have recognized that tearing off applets from the window permits the user the flexibility of moving applets to other locations on the screen, including inside a visible controlling applet. *See id.* As such, Banthia's "tearing off" feature suggests arranging selected applets within a visible controlling applet, such that each applet performs the same function, and yields no more than one would expect from such a combination. *See KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 417 (2007).

Appellants also assert that the Examiner has provided no reason to modify Banthia to display the controlling applet. App. Br. 20. Notably, Banthia at least suggests that the controlling applet can be visible (*see* FF 1), and thus, the Examiner is not suggesting modifying Banthia but rather

elected one of Banthia's alternative embodiments (e.g., atypical situation where the controlling applet is visible). Also, the Examiner explains (Ans. 13) that Figure 5 suggests to an ordinary skilled artisan displaying smaller, separate applets inside a larger portion of a window. FF 2. Moreover, the Examiner states (Ans. 13) an artisan would have recognized displaying the separate applets inside a controlling applet, in light of the controlling applet's working relationship with the other applets, such as interfacing and communicating with other applets to update applets with new data (FF 3). Lastly, as stated above, Banthia teaches a "tearing off" feature to move applets to various locations within the screen (*see* FF 4) and thus provides the user with flexibility in displaying selected applets inside the displayed main applet.

For the foregoing reasons, Appellants have not persuaded us of error in the rejection of independent claim 1 and claims 2-4, 9, 10, 14-18, 23-26, 28-32, 37-40, 42, 46-48, 51, 55, and 56 not separately argued with particularity (App. Br. 17-23).

Claim 11

Based on the record before us, we find no error in the Examiner's rejection of claim 11, which depends from claim 1. Notably, Appellants have not defined "a webtop applet" (*see generally* Specification) but have stated that "a windowing desktop within a browser[is] a 'WebTop'" (Spec. 9:16-17). As such, an applet which acts like a larger window or a desktop within a browser can reasonably be the recited webtop applet. As explained above when addressing claim 1, Banthia teaches and suggests the controlling or main applet can be visible such that other applets are inside of the main

applet or acts like a desktop within the browser. *See also* FF 1-4. Also, as the Examiner states (Ans. 5, 14), Banthia teaches applets, such as the controlling applet, are implemented or loaded into a Java interpreter (e.g., a Java application). *See* Banthia 2:30-36. For the foregoing reasons, Appellants have not persuaded us of error in the rejection of claim 11.

Claim 12

Based on Appellants' contentions, we find no error in the Examiner's rejection of claim 12, which depends from claim 1. We first refer to our previous discussion of claim 1 and how Banthia teaches and suggest a main applet displaying other applets within the main applet. Also, as the Examiner notes (Ans. 5, 13), Banthia's Figure 5 shows or displays various applets. *See* FF 2. Thus, Figure 5 is an index or roster (e.g., a list) of available applets. *See id.* While this list may differ from Appellants, this display is still a "list" given its broadest reasonable construction. Additionally, as discussed above, Banthia discloses how a user can move (e.g., select) an applet to another location on the screen. FF 4. Thus, each applet can be selected from this "list" as recited. For the foregoing reasons, Appellants have not persuaded us of error.

Claim 52

Claim 52 depends from claim 46. Based on the record before us, we are not persuaded by Appellants' arguments. As stated above when discussing claim 52 under § 112, first paragraph, the phrase, "base class" is not in or defined by the Specification. Appellants also have not demonstrated that a "base class" has a special meaning in the art. *See App.*

Br. 13-15, 24-25. Thus, the Examiner's construction that a base class includes "any parent class from which a derived class inherits from" (Ans. 14) is broad, but reasonable.

Banthia's Figure 3 shows a hierarchy or inheritance for various applets. Banthia 4:31-32; Fig. 3. As the Examiner explains (Ans. 7, 14), the "Packet Distribution Display" applet (as shown in Fig. 3) inherits the properties from the "Pie Chart" applet, or the Pie Chart is the base class from which the Packet Distribution Display applet inherits functions. *See id.* The Examiner also states the "Packet Rates Display" applet (also shown in Fig. 3) inherits the properties from the "Bar Chart" applet, or the Bar Chart is the base class from which the Packet Rate Display applet inherits functions. *See id.* We therefore agree with the Examiner that Banthia teaches a plurality of applets⁴ (e.g., the Packet Distribution Display and Packet Rates Display applets) that are independent in as much as each applet does not inherit the features or functions of its parent or base class (e.g., Pie Chart and Bar Chart). For the foregoing reasons, Appellants have not persuaded us of error in the rejection of claim 52.

Claim 54

Based on the record before us, we find no error in the Examiner's rejection of claim 54, which recites the receiving comprising receiving at substantially the same time the user selection of the applets. As discussed above in connection with claim 12, Banthia teaches and suggests displaying

⁴ Notably, claim 46, from which claim 52 depends, does not recite a "plurality of applets." Rather, claim 46 recites "a list of applets," "at least two applets," and "selected applets."

applets and the user selecting from the applets as a “torn off” window that can be moved from one location to another on the display screen. *See* FF 2, 4. Also, Appellants do not define “at substantially the same time” (*see generally* Specification) and, giving this phrase its broadest reasonable construction, “substantially the same time” can reasonably include being near in time (e.g., within several seconds of each other).

Banthia teaches tearing off multiple applets (Banthia 8:33-34) or selecting more than one applet. *See* Ans. 14. An ordinary skilled artisan would recognize from this teaching that multiple selections of different applets can occur subsequent to each other or adjacent in time. Thus, Banthia teaches and suggests selecting one applet to tear off and then immediately selecting another applet to tear off or “receiving at substantially the same time the user selection of the plurality of applets” as recited. For the foregoing reasons, Appellants have not persuaded us of error in the rejection of claim 54.

Claims 57, 58, and 60

Unlike claim 12, claim 57 recites the applets displayed in the list are not yet loaded for execution. We are therefore persuaded by Appellants’ arguments. While Figure 5 shows applets displayed and are arguably a displayed list of applets, these applets have been loaded for execution. *See* FF 2. The Examiner argues that the information in the applets is updated and thus recreates an applet that is not yet loaded. Ans. 9, 13. However, updating an applet with new information by redisplaying, as Banthia teaches (*see* Banthia 7:7-10), differs from loading an applet for execution. Also, Banthia’s Figure 5 does show various icons and displays various applets.

See FF 2. Nonetheless, whether these icons provide drop-down menus to select applets for display or how the applets displayed were uploaded in Figure 5 is not described in sufficient detail by Banthia.⁵ For the foregoing reasons, Appellants have persuaded us of error in the rejection of independent claim 57 and dependent claims 58 and 60 for similar reasons.

THE OBVIOUSNESS REJECTION OVER BANTHIA AND SOUTHGATE

Appellants refer to the discussion of independent claims 1, 15, 29, and 57 and additionally argue that Southgate does not cure the purported deficiencies. App. Br. 26. We disagree for the reasons previously discussed related to claim 1 and Banthia, and need not address whether Southgate cures any alleged deficiency. For the reasons previously set forth, Appellants have not persuaded us of error in the rejection of claims 5-8, 19-22, and 33-36.

However, claim 59 depends from claim 57. We stated above that Banthia does not teach or suggest the step of selecting applets from a displayed list of available applets, wherein the applets displayed in the list are not yet loaded for execution. Additionally, the Examiner has not relied upon Southgate to cure this deficiency. *See* Ans. 10-11. Because we reverse the Examiner's rejection of independent claim 57, we likewise will not sustain the rejection of claim 59.

⁵ Also, whether providing a menu or list to the user from the shown icons for selecting applets to be uploaded would have been obvious to an ordinary skilled artisan is not before us.

THE OBVIOUSNESS REJECTION OVER BANTHIA AND SUN

Regarding representative claim 13, Appellants refer to the discussion of independent claims 1, 15, and 29 and additionally argues that Sun does not cure the purported deficiencies. App. Br. 26-27. We disagree for the reasons previously discussed related to claim 1 and Banthia, and need not address whether Sun cures any purported deficiency. Additionally, Appellants argue that the Examiner provides no motivation or relies on impermissible hindsight in formulating the obviousness rejection. App. Br. 27-29. Appellants further assert that JInternalFrame is not intended for visibility, lacks operations for holding an applet, and thus does not suggest being implementing for applets. App. Br. 28. The issue before us then is: has the Examiner erred in rejecting claim 13 by finding that Banthia and Sun collectively would have taught or suggested generating a JInternal frame window for each selected applet under § 103?

ANALYSIS

Based on the record before us, we find no error in the Examiner's rejection of claim 13. We adopt the Examiner's findings related to Sun and the JInternal frame and combining this teaching with Banthia's desktop environment. *See* Ans. 11, 15. Particularly, the Examiner states Banthia teaches a desktop environment and Sun teaches using JInternal Frame components or objects inside a desktop (e.g., JDesktopPane). *See id.* Moreover, combining Sun's teaching of using a JInternalFrame with Banthia's desktop environment simply suggests arranging objects within Banthia's desktop (e.g., the selected applets), as a JInternalFrame, with each applet performing the same function, and yielding no more than one would

expect from such a combination. *See KSR*, 550 U.S. at 417. Thus, the Examiner provides an adequate explanation that the combination is obvious under § 103. *See id.* Additionally, Sun teaches using a JInternal Frame object to emulate a native frame window, such as Banthia's applet windows (*see* FF 4). *See* Sun 10. Thus, Sun provides yet another reason with some rational underpinning to support a legal conclusion for obviousness. *See KSR*, 550 U.S. at 418.

Lastly, Appellants assert that JInternalFrame is not intended for visibility, lacks operations for holding an applet, and thus does not suggest being implementing for applets. App. Br. 28. To support this assertion, Appellants refer to their own disclosure. *See id.* Yet, pages 2-4 and 9 do not state that the JInternalFrame lacks operations for holding an applet and thus cannot be implemented for applets. Also, while page 9 of the disclosure states JInternalFrame was not *intended* to have multiple applets visible (Spec. 9:12-13), this does not demonstrate that JInternalFrame cannot be generated for an applet or implemented to make the applets visible. Additionally, mere assertions that are unsupported by factual evidence are entitled to little probative value. *See Geisler*, 116 F.3d at 1470. For the foregoing reasons, Appellants have not persuaded us of error in the rejection of claim 13 and claims 27 and 41 not separately argued (App. Br. 26-29).

CONCLUSION

Under § 112, first paragraph, the Examiner did not err in rejecting claim 52.

Under § 103, the Examiner did not err in rejecting claims 1-42, 46-48, 51, 52, and 54-56 but erred in rejecting claims 57-60.

DECISION

The Examiner's decision rejecting claims 1-42, 46-48, 51, 52, and 54-60 is affirmed-in-part.

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv).

AFFIRMED-IN-PART

babc